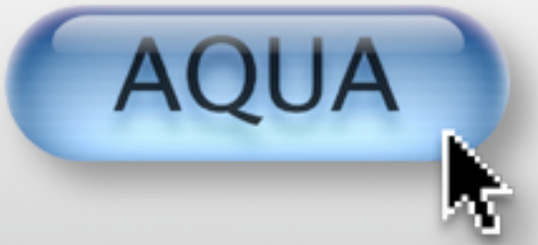
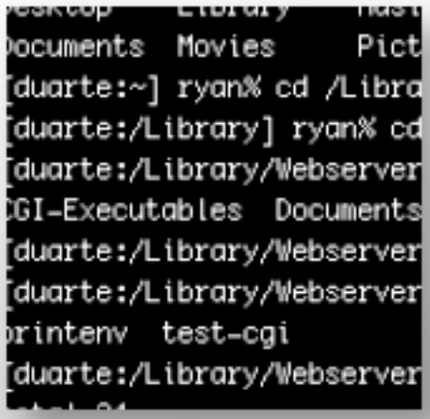
Carbon: The New HIToolbox

Session 409





Carbon: The New HIToolbox

Guy Fullerton
HIToolbox Engineer
guyf@apple.com

New HIToolbox?

What's the New HIToolbox?

- HIObject: Object model
- HIView: Composited view model
- Carbon Events: Messaging model
- Standard Event Handlers: Base-class behaviors

Built on a modern foundation

“To Stand Still Is to Regress”

Why Should You Care?

- New features are available through HIObject and HIView
- Best performance
- Easy, iterative adoption path
- Keep using portable C and C++ code

Take your Carbon app to the next level

What You Will Learn Today

- New views
  - HISearchField, HITextView, HISegmentView
  - HIMenuView, HIStandardMenuView
- New features
  - HITheme APIs
  - HIView layout
- Free stuff the New HIToolbox provides

Performance Improvements

- Event building, sending, and handling
- Window and view resize
- HIView manipulation
- Theme text drawing
- Menu item addition and manipulation
- DataBrowser add, remove, and select items

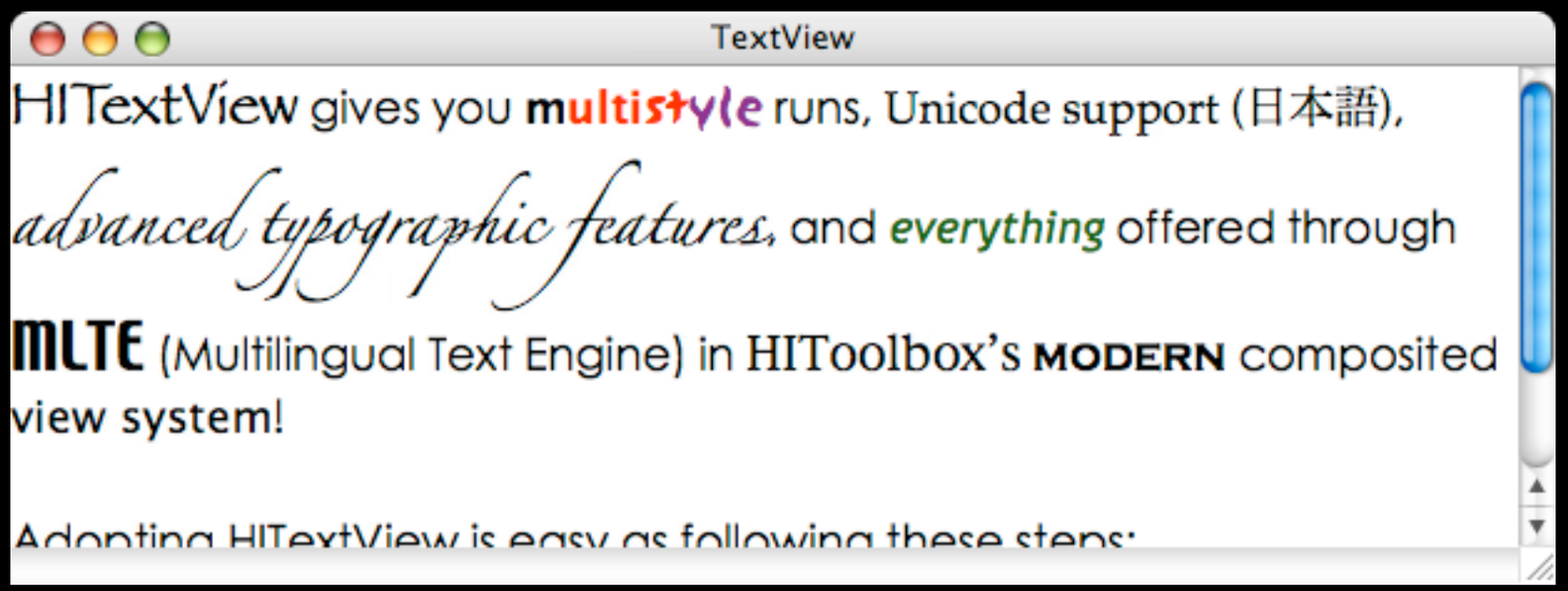
New in the Panther HIToolbox

- HITextView
- HISearchField
- HISegmentView
- HITheme APIs
- HIView Layout
- Custom Menu content via HIView
- Lots more…

HITextView

Why Use HITextView?

- Full-featured text editing HIView
  - Compositing mode only
- Implemented using MLTE
- Integrates with the Edit, Services menus
- Integrates with the Font Panel (soon)



HITextView Wraps Around MLTE

- HITextView API is very small
- Exposes the MLTE object
  - HITextViewGetTXNObject
- MLTE API is partially disabled for these TXNObjects
  - Can’t draw, change bounds, etc., with MLTE API
  - Use appropriate HIView APIs instead

HITextView and HIScrollView

- Designed to work together seamlessly
- Embed the HITextView in an HIScrollView
  - HITextView manages the text
  - HIScrollView manages the scrolling

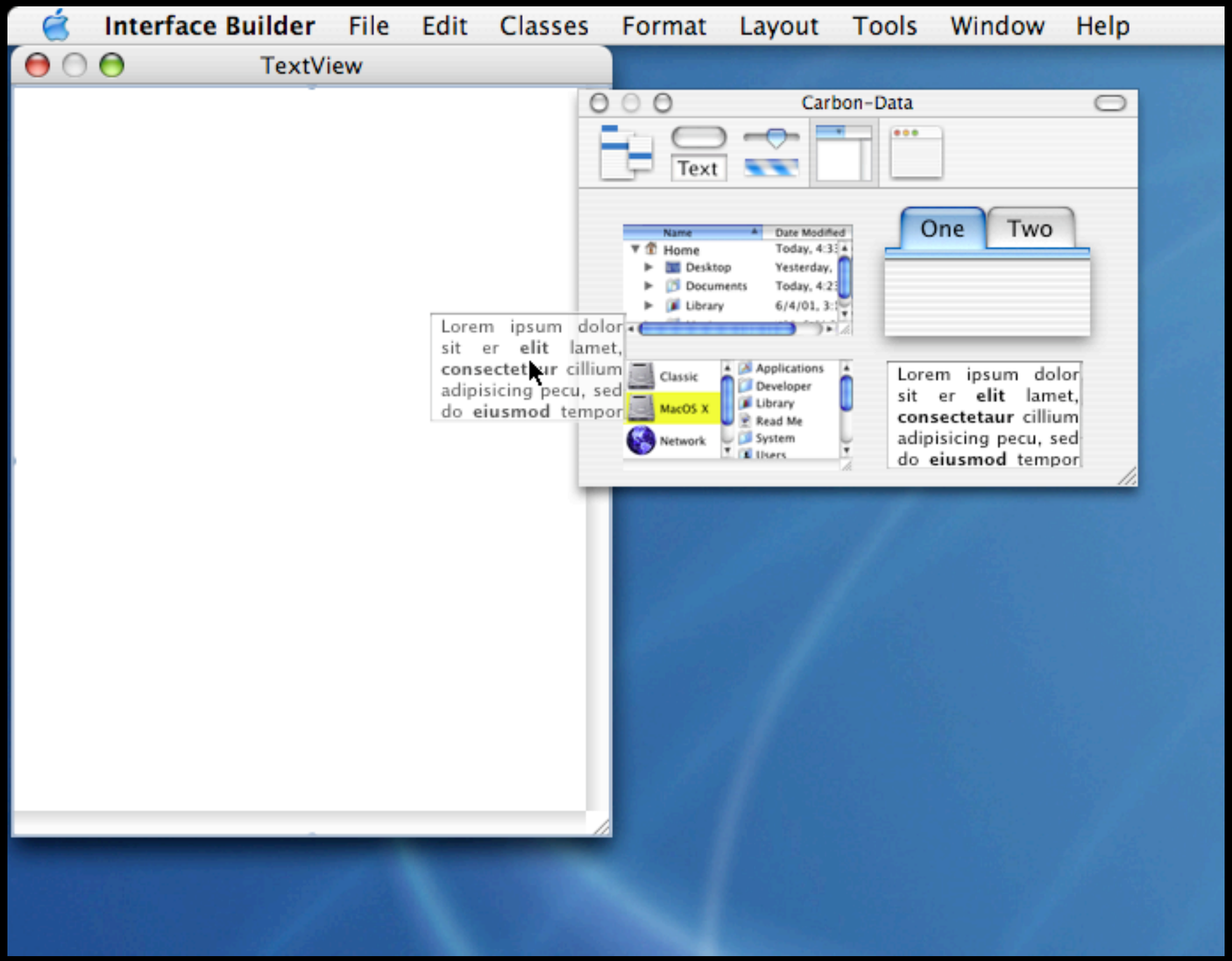
HITextView Creation

```
HIViewRef myTextView;

HITextViewCreate(   NULL, /* frame */
                    0, /* option bits */
                    0, /* TXNFrameOptions */,
                    &myTextView );

HIViewAddSubview( myScrollView, myTextView );
```

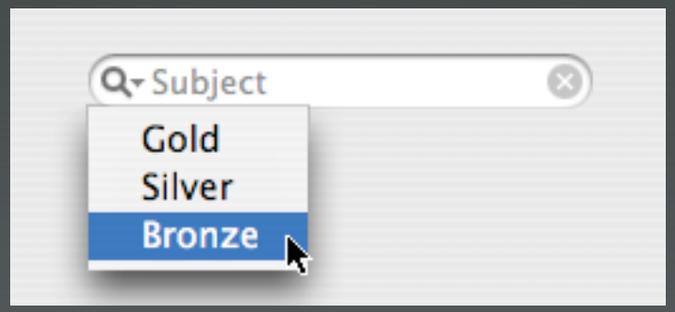
HITextView Manipulation

```
TXNObject   txn;
CFURLRef    myURL;

txn = HITextViewGetTXNObject( myTextView );

// ... build a URL to a file ...

TXNSetDataFromCFURLRef( txn, myURL,
    kTXNStartOffset, kTXNEndOffset );
```

Or...

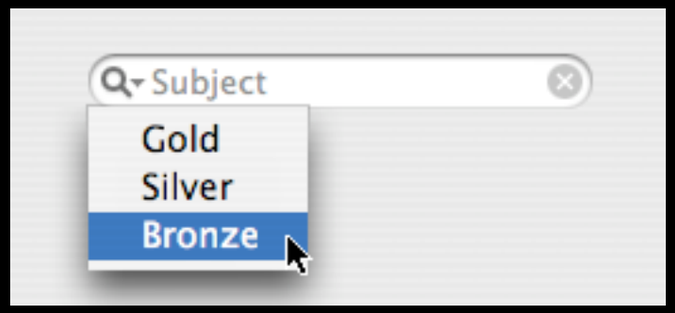

Interface Builder
File
Edit
Classes
Format
Layout
Tools
Window
Help
TextView
Carbon-Data
Text
Name
Date Modified
Home
Desktop
Documents
Library
Classic
MacOS X
Network
Applications
Developer
Library
Read Me
System
One
Two
Lorem ipsum dolor sit er elit lamet, consectetaur cillium adipisicing pecu, sed do eiusmod tempor
Lorem ipsum dolor sit er elit lamet, consectetaur cillium adipisicing pecu, sed do eiusmod tempor

HISearchField



Why Use HISearchField?

- Editable text field with rounded frame
- Optional cancel button
- Optional menu
- Optional descriptive label



HISearchField Creation

```
HIViewRef   mySearchField;
MenuRef     myMenu;

// ... create a MenuRef ...

HISearchFieldCreate(
    &frame, /* frame rect */
    kHISearchFieldAttributesCancel, /* attributes */
    myMenu, /* optional menu */
    CFSTR( “mailbox” ), /* descriptive text */
    &mySearchField );
```

HISearchField Creation

```
EventTypeSpec   events = {
    { kEventClassTextField, kEventTextAccepted },
    { kEventClassSearchField,
        kEventSearchFieldCancelClicked } };

InstallEventHandler(
    GetControlEventTarget( mySearchField ),
    MySearchFieldHandler,
    GetEventTypeCount( events ),
    events,
    mySearchField,  /* ref con */
    NULL );
```

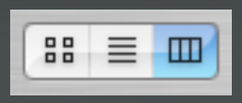
HISearchField Event Handling

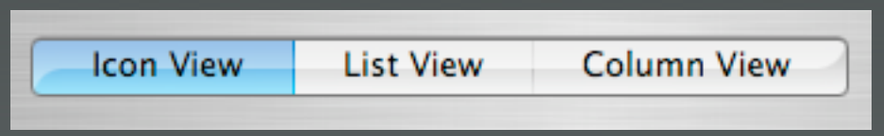
```
if ( class == kEventClassTextField &&
    kind == kEventTextAccepted )
{
    CFStringRef   string;
    GetControlData( searchField, kControlNoPart,
        kControlEditTextCFStringTag,
        sizeof( string ), NULL, &string );

    MyDoSearch( searchField, string );

    CFRelease( string );
}
```

HISearchField Event Handling

```
else if ( class == kEventClassSearchField &&
    kind == kEventSearchFieldCancelClicked )
{
    MyCancelSearch();
}
```

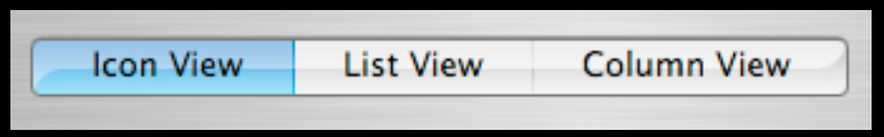
HISegmentView



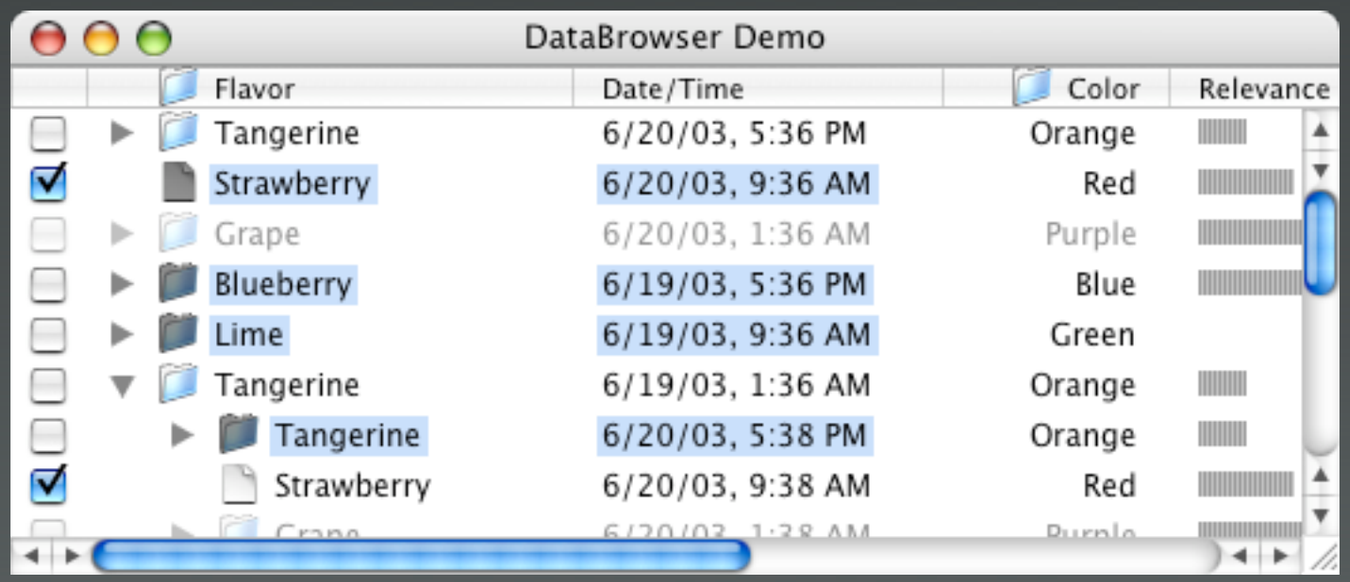


HISegmentView

- Radio, Momentary, and Sticky behaviors
- Customizable content
- Not quite ready for the WWDC build
  - Will be ready for Panther GM



Data Browser Improvements



DataBrowser in Compositing Mode

- It works!
- More efficient in compositing mode
- Work in progress
  - Feedback is appreciated!

List View Type-Selection

- Specify your type-select column with a new property:
  - kDataBrowserListViewTypeSelectColumn
- Type-selection keyboard input via Carbon Events

HITheme

Drawing Primitives

Why Adopt HITheme APIs?

- Much faster than Appearance Manager APIs
- Still familiar usage model
- Draw pulsing push buttons!
- Primitives can draw the newest Aqua widgets
- Only way to print via theme drawing primitives

HITheme vs. Appearance Manager

- Completely Quartz-savvy
- Uses HIPoint and HIRect parameters
- Expect the familiar top-left origin
  - Works with bottom-left origin, too

HIView Layout

Rebounder Demo

Why Use HIView Layouts?

- Layouts manage complex spacial relationships
  - Far more than just springs & rods or bindings
- Toolbox can layout at the most efficient times
- Eliminates or reduces your app's layout code
- Details in Session 425: HIView in Depth

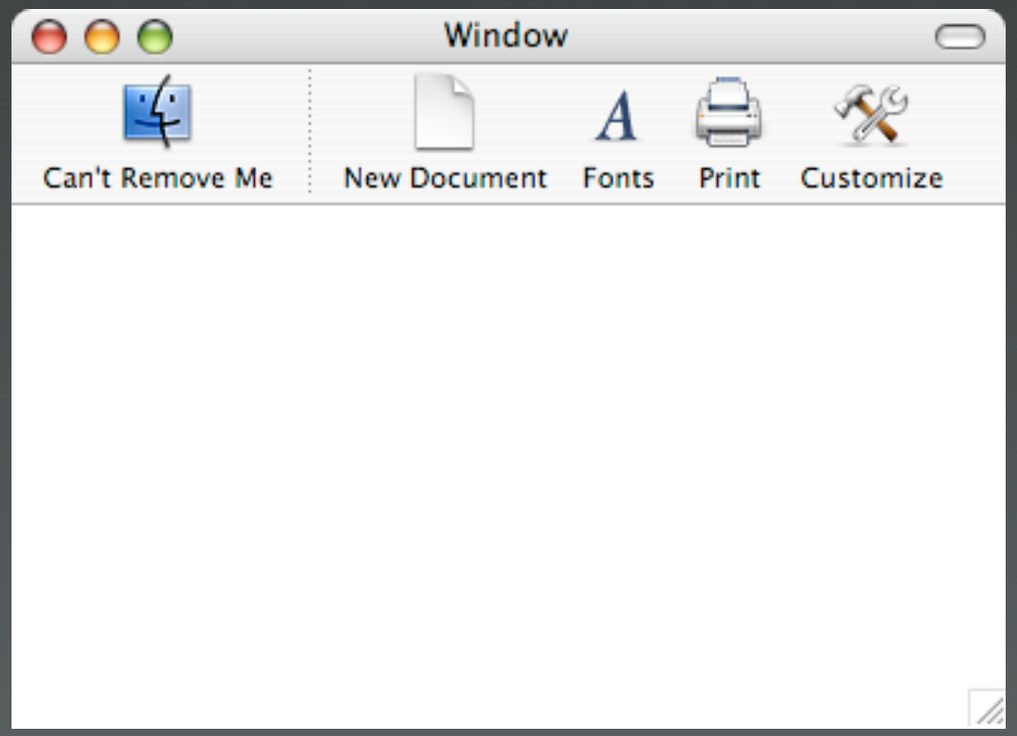
HIAboutBox

SuperUltraMega Pro
2.1
The most super, most ultra, and most mega professional application to date!
Copyright 2003, All Rights Reserved

Why Use HIAboutBox?

- Gives you the HI-conformant about box look
- Near zero effort to adopt
- Fetches info from your App Bundle or “BNDL”
  - You can customize if you wish
- Displays in response to kHICommandAbout
  - You can display it manually, too
- Dismisses in response to kHICommandClose

HIToolbar Additions



HIToolbar Additions

- HIToolbarItemConfigDataChanged
  - Custom items report changed persistent data
- HIToolbarSetItemsWithIdentifiers
  - Most efficient way to manually populate a toolbar

Carbon Event Additions

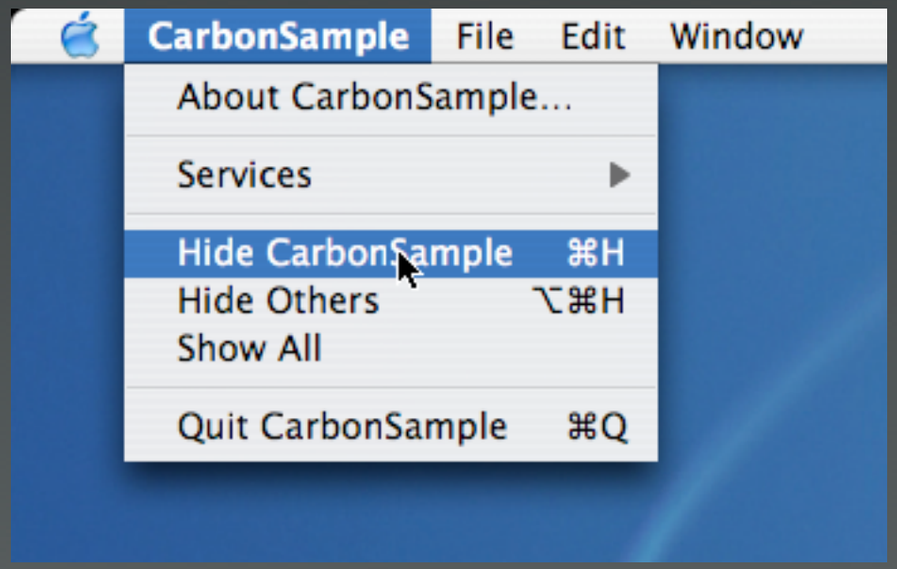
Carbon Event Additions

- HIMouseTrackingGetParameters
- CopyEventAs
- AcquireFirstMatchingEventInQueue
  - Modern replacement for EventAvail and StillDown
  - Does not call the event loop or flush

GetEventMonitorTarget

- Observe other Apps' user input!
  - Key down/up, mouse down/up
- Can't see secure input
- Very performance intensive

Menu Manager Additions



Menu Manager Additions

- CancelMenuTracking
- kEventMenuBarShown/Hidden
  - Sent to the Application target

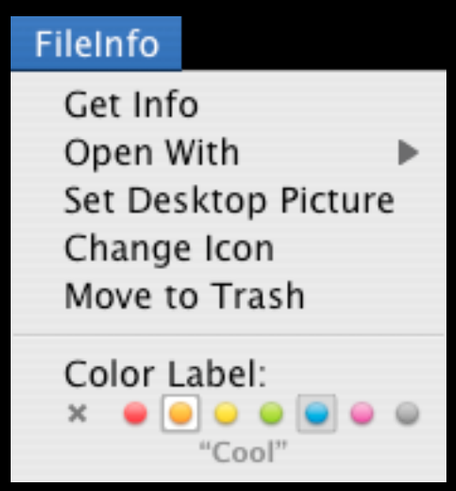
HIView-based Menus

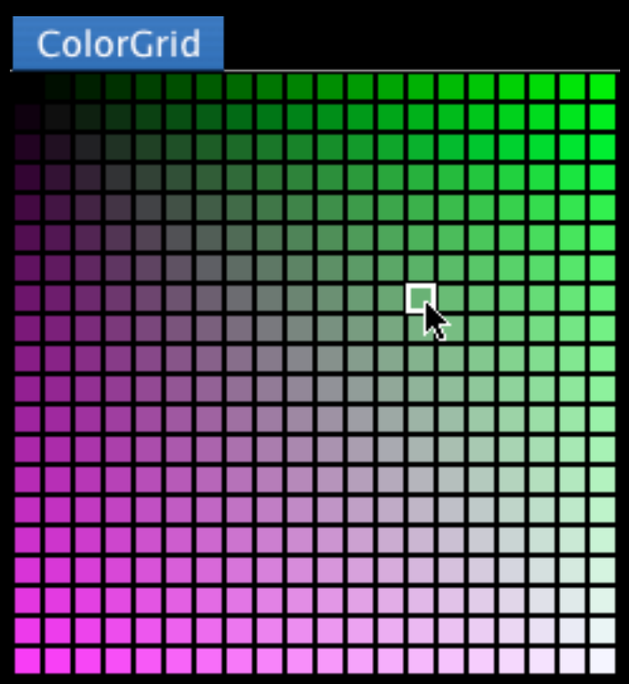
Menu View Basics

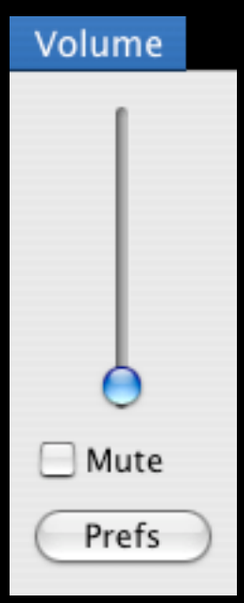
- Menus are drawn in compositing windows
- Menu Content View draws the menu items
  - Standard view draws text, icons, command keys
- Menu Window Frame View draws the background
- HIMenuGetContentView
  - Gets the view that draws the menu items for a menu
- HIMenuViewGetMenu
  - Gets the MenuRef that is drawn by a menu content view

Why Make a Custom Menu?

- Menus that aren't just text and icons
- Grid or X,Y menus: tool or color palette
- "Menus" that aren't really menus: Volume controller







How to Make a Custom Menu

- Use HIObject to inherit from kHIMenuViewClassID
- Embed other HIViews as desired
- Handle some standard HIView Carbon Events
  - View part codes are the menu item indices

Required Carbon Events

- kEventControlDraw
  - Draw your menu items
  - Focused item should be drawn selected
    - HIViewGetFocusPart
- kEventControlHitTest
  - Report the menu item under the point
- kEventControlGetPartRegion
  - Report the region of the given menu item

More Required Carbon Events

- kEventControlGetOptimalBounds
  - Calculate ideal size of the menu content
  - Menu may be sized smaller
  - Scrolling will be handled for you...
- kEventScrollableGetInfo
  - Provide some scrolling details

Other Carbon Events

- Default behavior provided by kHIMenuViewClassID
- kEventMenuCreateFrameView
  - Create a view to draw a custom frame & background
- kEventMenuGetFrameBounds
  - Determine global screen position of the menu
- kEventMenuBecomeScrollable
  - Block/override default menu scrolling functionality
  - Corresponding kEventMenuCeaseToBeScrollable

Other Techniques

- Override standard menu view
  - kHIStandardMenuViewClassID
- Use an HIView to represent an individual menu item
- Put other HIViews in the menu window
- Sample code available

Window Manager Additions

Window Manager Additions

- HIWindowChangeClass
  - Document window <-> Floating window
- HIWindowFlush
- HIWindowIsDocumentModalTarget
  - Detect whether a sheet is being displayed
- DetachSheetWindow
- kEventAppActiveWindowChanged
- kEventAppAvailableWindowBoundsChanged

Window Transitions

- kWindowGenieTransitionEffect
  - Genie whenever and wherever you want!
- TransitionWindowWithOptions
  - Customize transition durations
  - Support for asynchronous transitions
  - Provide user data to the new Carbon Events…
- kEventWindowTransitionStarted
- kEventWindowTransitionCompleted

Asynchronous Window Dragging

- Better responsiveness
  - Improves perceived performance
- Allows user to drag windows of hung/busy apps
- Easy to turn on:
  - kWindowAsyncDragAttribute
- Won’t receive kEventWindowBoundsChanging events

What You Can Do Now

- Use the new Aqua widgets
- Work toward adopting HIView compositing
- Use HITheme APIs instead of the Appearance Manager

Roadmap

415 Data Exchange Options for Carbon Applications	Russian Hill Thu., 10:30am
425 Carbon: HIView in Depth	Presidio Fri., 3:30pm
416 Carbon: Optimizing Performance	Presidio Thu., 2:00pm
307 Using Apple Tools for Carbon Development	Russian Hill Thu., 9:00am

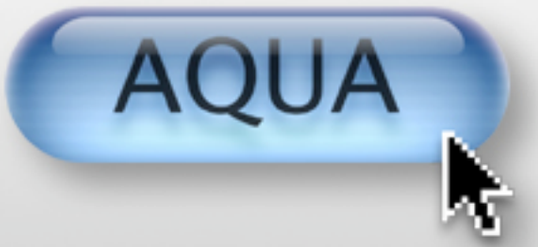
Who to Contact

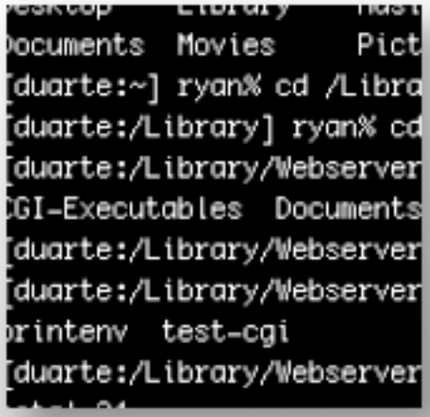
Xavier Legros
That French guy
xavier@apple.com

John Geleynse
Manager, Software Evangelism
geleynse@apple.com

http://developer.apple.com/wwdc2003/urls.html

Q&A





Guy Fullerton
HIToolbox Engineer
guyf@apple.com

http://developer.apple.com/wwdc2003/urls.html

WWDC 2003

Worldwide Developers Conference